А. Н. Дробахина

ОСНОВЫ ФРАКТАЛЬНОЙ ГРАФИКИ

Понятие компьютерной графики не ограничивается такими ее видами как растровая и векторная. В некоторых учебных пособиях по информатике рассматривается третий вид компьютерной графики – фрактальная [11].

Термином «фрактал» (от латинского fractus – дробленый, состоящий из фрагментов) обозначают геометрическую фигуру, обладающую свойством самоподобия, то есть составленную из нескольких частей, каждая из которых подобна всей фигуре целиком. Этот термин был предложен французским математиком Бенуа Мандельбротом в 1975 году и получил распространение с выходом в 1977 году его книги «Фрактальная геометрия природы» [9].

В настоящее время возрастает интерес к применению методов фрактальной геометрии в различных областях науки и техники – физике, астрономии, биологии, медицине, социологии, экономике, информационных и телекоммуникационных технологиях и др. Так, в физике фракталы применяются при изучении поглощения или рассеяния излучения в пористых средах, при анализе процессов усталостного разрушения материалов, для характеристики сильно развитой турбулентности, при моделировании свойств поверхности твердых тел, для описания диэлектрического пробоя и молнии и пр. [2]. В биологии фракталы применяются, например, для моделирования процессов, происходящих во внутренних органах (процессов кровообращения, биения сердца и др.), моделирования популяций. Фракталы используются и для моделирования поведения хаотических динамических систем, таких, как поведение погоды, описан способ моделирования исторических процессов и явлений средствами фрактальной геометрии [7]. В экономике фракталы применяют при анализе колебаний курса валют [12].

Широко применяются фракталы и в области компьютерной графики [2, 3, 14], например, для получения изображений природных объектов (растений, облаков, гор, береговых линий, ландшафтов, поверхности морей, карт и пр.) при создании компьютерных игр. Другое перспективное направление – фрактальное сжатие изображений [12], позволяющее добиться сверхвысокой степени сжатия (по некоторым данным [10]– до 2000 раз) с малыми потерями качества.

Фрактальная графика, как векторная, является вычисляемой. Но, в отличие от векторной графики, основное внимание уделяется не заданию изображения в виде совокупности простых геометрических фигур (прямых, многоугольников, окружностей, многогранников и пр.), а описанию алгоритма построения изображения – фрактальное изображение строится по уравнению или системе уравнений, и никакие объекты в памяти компьютера не хранятся.

Существует несколько классификаций фракталов [2, 3, 14]. В рамках одной из них [3] выделяют геометрические, алгебраические и стохастические фракталы.

Геометрические фракталы. История фракталов началась с геометрических фракталов, которые исследовались математиками в XIX веке. Фракталы этого класса самые наглядные, в них сразу видна самоподобность частей.

Примерами геометрических фракталов являются: треугольник Серпинского, триадная кривая Кох, кривая Пеано, снежинка Кох, кривая дракона, губка Менгера и др.

Геометрические фракталы получаются путем несложных геометрических построений [1]:

1. задается начальное условие (нулевое поколение): фигура, на основании которой строится фрактал;

2. задается процедура (генератор), которая определенным образом преобразует нулевое поколение;

3. в результате бесконечного повторения заданной процедуры получается геометрический фрактал.

На рисунке 1 изображен геометрический фрактал салфетка Серпинского.

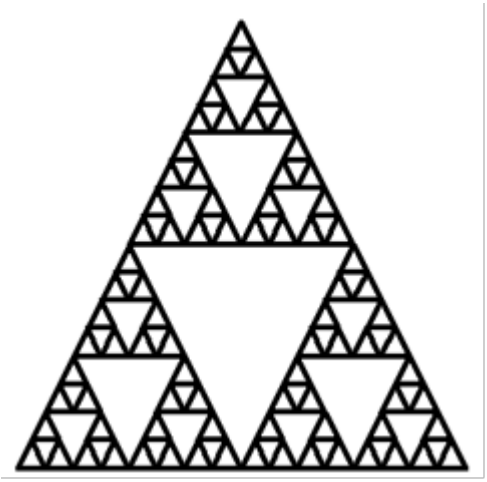


Рис 1. Фрактал салфетка Серпинского

Алгоритм построения салфетки Серпинского следующий:

1. задаем начальное условие: нулевое поколение представляет собой правильный треугольник со стороной 1;
2. определяем процедуру, которая преобразует нулевое поколение: делим средними линиями исходный треугольник на четыре равных треугольника и центральный выбрасываем. Получаем три треугольника со стороной ½;
3. повторяем эту процедуру с тремя оставшимися треугольниками требуемое количество раз.

Один из вариантов реализации рассмотренного алгоритма в Delphi представлен ниже.

```
Procedure DrawTriangle (x1, y1, x2, y2, x3, y3: integer);
```

```
Begin
      form1.Canvas.Polygon ([Point(x1, y1), Point(x2, y2), Point(x3, y3)]);
End;
Procedure DrawGenerator(x1, y1, x2, y2, x3, y3, n: integer);
Var x1n, y1n, x2n, y2n, x3n, y3n : integer;
Begin
    If n > 0 then
Begin
x1n := (x1 + x2) div 2;
y1n := (y1 + y2) div 2;
x2n := (x2 + x3) div 2;
y2n := (y2 + y3) div 2;
 x3n := (x3 + x1) div 2;
 y3n := (y3 + y1) div 2;
 DrawTriangle(x1n, y1n, x2n, y2n, x3n, y3n);
 DrawGenerator(x1, y1, x1n, y1n, x3n, y3n, n - 1);
 DrawGenerator(x2, y2, x1n, y1n, x2n, y2n, n - 1);
 DrawGenerator(x3, y3, x2n, y2n, x3n, y3n, n - 1);
End;
End;
procedure TForm1.Button1Click(Sender: TObject);
var n, c, x, y: integer;
begin
val(inputbox('Введите количество вложений', 'n', '4'),n,c);
if (c=0) then
begin
form1.brush.Color:=form1.Color;
form1.canvas.FillRect (clientrect);
```

```
x:=form1.ClientWidth;

y:= form1.ClientHeight;

If x>y then x:=y else y:=x;

DrawTriangle(10,y-10,x div 2,10,x-10,y-10);

DrawGenerator(10,y-10,x div 2,10,x-10,y-10,n);

end;

end;
```

Для выполнения построений используется две процедуры: первая – DrawTriangle, строящая треугольник по заданным точкам, а вторая – рекурсивная процедура DrawGenerator, выполняющая пересчет точек – вершин треугольников и трижды вызывающая себя в этих точках. В обращении к процедуре включен целочисленный аргумент n, определяющий глубину рекурсии (количество вложений).

Вторая группа фракталов – ***алгебраические фракталы***. Свое название они получили потому, что их строят на основе алгебраических формул.

Классический пример алгебраических фракталов – множество Мандельброта, описанное французским математиком Пьером Фату еще в 1905 году. Однако впервые оно было построено Бенуа Мандельбротом в 1980 г.

При построении множества Мандельброта наиболее часто используется функция комплексного переменного:

$$z_{i+1}=z_i^2+c \qquad (1)$$

где Z и C – комплексные числа, а Z0=0.

Раскладывая уравнение на действительную и мнимую часть, эту формулу заменяют двумя выражениями:

$$\begin{aligned} x_{i+1}&=x_i^2-y_i^2+a \\ y_{i+1}&=2\cdot x_i\cdot y_i+b \end{aligned} \qquad (2)$$

где X0 и Y0 для каждой новой точки изначально равны нулю.

При построении этого фрактала используется теорема: если точка удалится от начала координат на две и более единицы, то она уйдет в бесконечность и окажется за пределами множества Мандельброта. Эта теорема служит критерием определения принадлежности точки множеству Мандельброта, и фактически является способом его построения.

Поэтому для каждой точки с координатами (a,b) проводится серия вычислений: кроме расчета значений Xi+1 и Yi+1 по формуле (2), на каждом шаге вычисляют величину $r_i = \sqrt{x_i^2 + y_i^2}$, представляющую собой расстояние от точки (Xi, Yi) до начала координат. Если ri <2 при любом числе итераций, то точка с координатами (a,b) принадлежит множеству Мандельброта, и окрашивается в черный цвет. Если же при некотором значении i точка вышла за пределы окружности, то считается, что точка не принадлежит множеству Мандельброта и на этом итерационный процесс для нее завершается. Сама точка при этом окрашивается в белый цвет.

Очевидно, что реально продолжать итерационный процесс до бесконечности невозможно, и на практике ограничиваются неким конечным, но достаточно большим числом. Известно, что для достижения приемлемой точности достаточно 100 итераций [8].

Полное изображение множества Мандельброта строится на участке координатной плоскости от -2 до 1 по оси X, и от -1,5 до 1,5 - по оси У.

Так как точка либо принадлежит множеству Мандельброта, либо нет, то его изображение должно быть черно-белым. Однако, учитывая то, что точки, не принадлежащие множеству Мандельброта, покидают окружность с центром в начале координат и радиусом 2 за разное количество шагов, можно построить цветные изображения.

Наиболее распространен способ построения цветного изображения множества Мандельброта, при котором точки, принадлежащие множеству, окрашиваются в черный цвет, а не принадлежащие множеству окрашиваются в цвет, равный количеству итераций, за которое точка покидает эту окружность. Причем выбранный цвет может быть не каким-либо конкретным значением, а использоваться как аргумент какой-либо функции выбора цвета. Меняя параметры, можно получить различные узоры.

Существует большое количество алгоритмов для построения множества Мандельброта [2, 5, 8, 12]. Ниже представлен фрагмент программы, реализующей один из подходов к построению множества Мандельброта.

```
function DrawMandelbrot(a, b: real):boolean;

var x, y, r, t: real;

Iter: integer;

begin

x:=0; y:=0; r:=0; Iter:=0;

while (Iter<=100) and (r<4) do

begin

t:=x;
```

```
x:=x*x-y*y+a;
y:=2*t*y+b;
r:=x*x+y*y;
inc(Iter);
end;
if x*x+y*y<4 then DrawMandelbrot:= true else DrawMandelbrot:=false;
end;
procedure TForm1.Button1Click(Sender: TObject);
var i, j : Integer;
 x_min, x_max, y_min, y_max, x, y, dX, dY: real;
 begin
 x_min := -2.0;
 y_min := -1.5;
 x_max := 1.0;
 y_max := 1.5;
 dx := (x_max - x_min) / PaintBox1.ClientWidth;
 dy := (y_max - y_min) / PaintBox1.ClientHeight ;
 y := y_min;
for j := 0 to PaintBox1.ClientHeight do
begin
 x := x_min;
 for i := 0 to PaintBox1.ClientWidth do
begin
if DrawMandelbrot (x, y) then PaintBox1.Canvas.Pixels [i,j]:=clblack
else PaintBox1.Canvas.Pixels [i,j]:=clwhite;
x := x + dx;
end;
y:= y + dy;
```

```
end;

end;
```

В данной программе логическая функция DrawMandelbrot определяет принадлежность точки (a,b) множеству Мандельброта: точка, принадлежащая множеству, окрашивается в черный цвет, а не принадлежащая – в белый (рисунок 4).

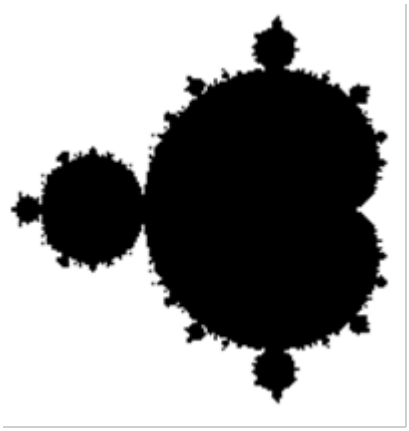


Рис. 4. Множество Мандельброта

Стохастические фракталы. Еще одним известным классом фракталов являются стохастические фракталы, которые получаются в том случае, если в процессе построения случайным образом менять какие-либо его параметры. При этом получаются объекты очень похожие на природные – несимметричные деревья, изрезанные береговые линии и т.д. Именно двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря [2, 14].

Примерами стохастических фракталов являются траектория частицы, совершающей броуновское движение, различные виды рандомизированных фракталов, то есть фракталов, полученных с помощью рекурсивной процедуры, в которую на каждом шаге введен случайный параметр.

В качестве примера стохастического фрактала рассмотрим построение фрактального дерева, алгоритм и реализация которого предложены в [5].

Автор предлагает следующий алгоритм построения фрактального дерева. Сначала строится ствол дерева случайной длины, от него строятся несколько ветвей также случайной длины, при этом их толщина уменьшается. Затем от каждой из веток в зависимости от их длин может строиться еще несколько веток, и цикл повторяется. На каждом шаге проверяется длина ветки: если она меньше некоторой заранее определенной величины, то вместо веток рисуется лист, и для этой ветки итерационный процесс прекращается.

Фрактальное дерево представлено на рисунке 5.

Рис. 5. Фрактальное дерево

```
procedure Tree(x, y, l: Integer; a: Real);
var x1, y1, p, s, i : Integer;
a1 : Real;
begin
 if l>25 then
 begin
 x1 := trunc(x + l*cos(a));
 y1 := trunc(y + l*sin(a));
 if l > 100 then p:=100 else p:=l;
 if p > 50 then
 begin
 for i:=0 to (p div 10) do
 begin
 Form1.Canvas.Pen.Color:=clBlack;
 Form1.Canvas.Polygon([Point(x+i-(p div 20), y),Point( x1, y1)]);
 end;
 end
else
begin
```

```
if Random > 0.5 then Form1.Canvas.Pen.Color:=clgreen

else Form1.Canvas.Pen.Color:=clolive;

 for i:=0 To 3 do Form1.Canvas.Polygon([Point(x+i, y),Point( x1, y1)]) ;

 end ;

for i:=0 to 10-Random(10) do

 begin

 s := Random(l-l div 5) + (l div 5);

 a1:= a + pi/2*(0.5-Random);

 x1:= trunc(x + s*cos(a));

 y1:= trunc(y + s*sin(a));

 Tree(x1, y1,p-10-Random(10), a1);

 end;

 end;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

 Randomize;

 form1.Canvas.FillRect(clientrect);

 Tree(Form1.ClientWidth div 2, Form1.ClientHeight-10, Form1.ClientHeight div 3,-pi/2 );

 end;
```

Как видно, создание фрактального изображения состоит в программировании, а не рисовании, поэтому фрактальную графику лучше всего изучать на примере какого-либо языка программирования. Однако с фрактальной графикой студентов можно ознакомить при изучении дисциплины «Программное обеспечение ЭВМ». Причем сделать это можно в разделе «Системы машинной графики», изучая, например, возможности графического редактора Gimp.

Рассмотрим эти возможности [4].

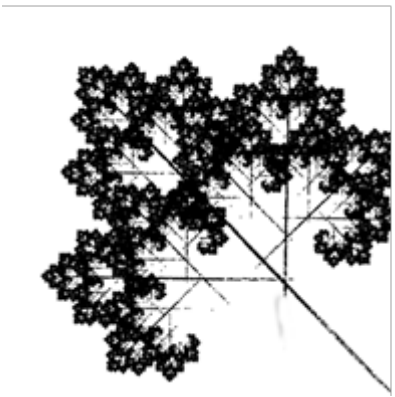
Для создания фрактальных изображений в Gimp встроены специальные фильтры:

1. Фильтр – Визуализация – Исследователь фракталов,
2. Фильтр – Визуализация – Природа (Пламя и IFT-фрактал);

3. Фильтр – Карта – Фрактальный след.

Создадим фрактальное изображение с помощью Исследователя фракталов.

1. Создайте новый документ размером 400х400 пикселей;
2. Примените фильтр ***Исследователь фракталов***. Окно Исследователя фракталов содержит две панели: слева находится панель предварительного просмотра с настройкой масштаба, справа расположены главные настройки, разбитые на закладки **Параметры**, **Цвета** и **Фракталы**.Во вкладке **Фракталы** можно выбрать форму фрактала, основной его вид. Во вкладке **Параметры** форму фрактала можно видоизменить, подвигав ползунки и выбрав тип фрактала. Во вкладке **Цвета** – изменить настройки цвета изображения. В области предпросмотра можно выделить часть изображения (протаскивая мышь с нажатой левой кнопкой). После окончания выделения выбранная часть заполняет собой всю область предпросмотра, а после нажатия на кнопку «ОК» содержимое области предпросмотра появляется в окне изображения. Выберите фрактальное изображение, измените его при помощи вкладок **Параметры и Цвета, выберите** понравившуюся область, и нажмите ОК.
3. Усложните фрактальное изображение, применив фильтр Фрактальный след. Теперь изображение можно сохранить.



Создадим фрактальное изображение (лист) с помощью фильтра IFS.

Рис. 6. Фрактальное изображение листа

IFS означает "Iterated Function System" − повторяемые функциональные системы. При помощи этого инструмента вы можете создать такие формы, как листья, цветы, ветки или деревья.

- Создайте новый файл;
- Установите цвет переднего плана в панели инструментов: чёрный, а цвет фона — белый.
- Откройте "Фильтры" – "Визуализация" –"Природа" – "IFS-фрактал".

Важно помнить, что результат трудно предсказать, поэтому нужно очень осторожно менять структуру. Если треугольный компонент слишком большой или сдвинут слишком далеко, то окно просмотра станет сплошным чёрным или изобразит бесформенное облако частиц. Если вы нашли желаемую текстуру(смотрите на результат в окне просмотра справа), делайте небольшие изменения.

- Выделите треугольник № 3 и поверните его на 75°.
- Поверните треугольник № 2 на 55°.

Получились контуры кончика и краёв будущего листа.

- Создайте новый элемент — "стебель", нажав значок Создать на панели инструментов.
- Растяните новый элемент — "стебель" (значок Растянуть) уменьшив масштаб до 0,1, и расположите его таким образом, чтобы получились подобие листа

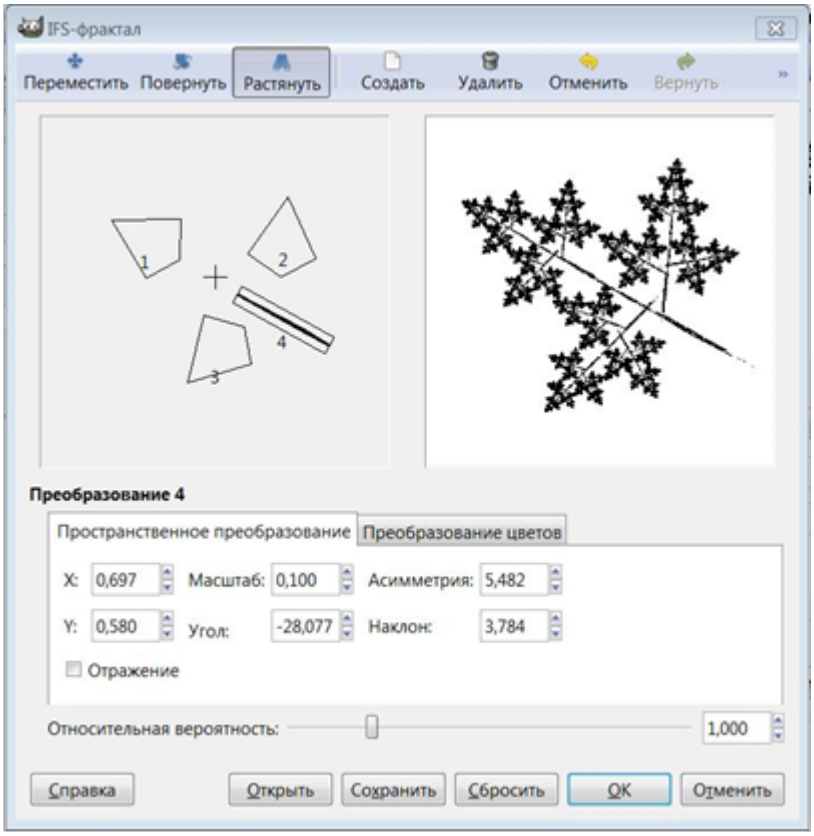


Рис. 7.Окно фильтра IFS-фрактал

- Измените форму и масштабы элементов 1—3 для получения требуемой формы.
- Сгруппировав все объекты, поверните их до получения требуемого изображения (смотрите на результат в окне просмотра справа).
- Преобразуйте цвета, задав каждому элементу свое значение цвета.
- Нажмите кнопку "OK" чтобы применить изображение и получить фрактальный лист. Теперь результат можно сохранить.

Литература

1. Алгебраические фракталы. [Электронный ресурс] / В.А. Трухачева, И. Сидоренко, А.В. Бородина. – Электрон. текстовые данные (32998 bytes). – Режим доступа: http://dssp.petrsu.ru/~KOF/kse-pact/index.html
2. Божокин С.В. Фракталы и мультифракталы: Учебное пособие/ С.В. Божокин, Д.А. Паршин – Москва–Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. – 128 с.
3. Ватолин Д.С. Применение фракталов в машинной графике. / Д.С. Ватолин // Computerworld-Россия. – 1995. – N15. – С. 11
4. Вольхин К.А. Основы компьютерной графики: Электронные методические указания к лабораторным работам для студентов [Электронный ресурс] / Вольхин К.А. http://grafika.stu.ru/wolchin/umm/pr_kg/
5. Генерация фрактальных деревьев [Электронный ресурс] / М.В. Котов. – Электрон. текстовые данные (6372 bytes). – Режим доступа: http://fractalworld.narod.ru/article/tree3.html
6. Доступно о фракталах [Электронный ресурс] – Электрон. текстовые данные (6242 bytes). – Режим доступа: http://fract.narod.ru/index.htm
7. Жуков Д.С. Моделирование исторических явлений и процессов средствами фрактальной геометрии. [Электронный ресурс] / Д.С. Жуков, С.К. Лямин. –– Электрон. текстовые данные (33888 bytes). – Режим доступа: http://www.ineternum.ru/ineternum/fraktal/vist_konf_frakt/istoriia_i_komp/8.htm
8. Колесников А. Визуализация фрактальных структур. [Электронный ресурс] /А. Колесников // Азбука программирования. – 1999. – N 16. – Электрон. текстовые данные (56786 bytes). – Режим доступа: http://www.kv.by/index1999161201.htm
9. Мандельброт Б. Фрактальная геометрия природы. / Б. Мандельброт. – М.: Институт компьютерных исследований, 2002. – 656 с.
10. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин – М.: Диалог-МИФИ, 2002. – 384 с.
11. Могилев А.В. Информатика: Учеб. пособие для студ. пед. вузов / А.В. Могилев, Н.И. Пак, Е.К. Хеннер; Под ред. Е.К. Хеннера. – 3-е изд., перераб. и доп. – М.: Академия, 2004. – 848 с.
12. Морозов А.Д. Введение в теорию фракталов. / А.Д. Морозов. – Москва–Ижевск: Институт компьютерных исследований, 2002. – 160 с.
13. Спиридонов Ф.Ф. Случайные фракталы: реализация в Maple / Ф.Ф.Спиридонов, В.В. Смирнов // Exponenta Pro. Математика в приложениях. – 2004. – №.#3–4 (7–8). – С. 138 – 141.
14. Шабаршин А.А. Введение во фракталы. [Электронный ресурс] / А.А. Шабаршин. –– Электрон. текстовые данные (30902 bytes). – Режим доступа: http://algolist.manual.ru/graphics/fracart.php